

**Pencarian SPBU Terdekat
Menggunakan *Finite State Automata*
(Studi Kasus: SPBU Wilayah Salatiga)**

Artikel Ilmiah



**Peneliti :
Diah Ayu Wulandari (672013137)
Magdalena A. Ineke Pakereng, M.Kom.**

**Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Salatiga
November 2016**

**Pencarian SPBU Terdekat
Menggunakan *Finite State Automata*
(Studi Kasus: SPBU Wilayah Salatiga)**

Artikel Ilmiah

**Diajukan kepada
Fakultas Teknologi Informasi
untuk memperoleh Gelar Sarjana Komputer**



1956

Peneliti :

**Diah Ayu Wulandari (672013137)
Magdalena A. Ineke Pakereng, M.Kom.**

**Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Salatiga
November 2016**



PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : DIAN AYU WULANDARI
NIM : 672013137 Email : AYUD570@GMAIL.COM
Fakultas : FTI Program Studi : TEKNIK INFORMATIKA
Judul tugas akhir : PENCARIAN SPBU TERDEKAT MENGEUNAKAN FINITE STATE
AUTOMATA (STUDI KASUS : SPBU WILAYAH SALATIGA)
Pembimbing : 1. MAGDALENA A. INEKE PAKERENG, M.Kom
2. _____

Dengan ini menyatakan bahwa:

1. Hasil karya yang saya serahkan ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar kesarjanaan baik di Universitas Kristen Satya Wacana maupun di institusi pendidikan lainnya.
2. Hasil karya saya ini bukan saduran/terjemahan melainkan merupakan gagasan, rumusan, dan hasil pelaksanaan penelitian/implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan narasumber penelitian.
3. Hasil karya saya ini merupakan hasil revisi terakhir setelah diujikan yang telah diketahui dan disetujui oleh pembimbing.
4. Dalam karya saya ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali yang digunakan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari terbukti ada penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya saya ini, serta sanksi lain yang sesuai dengan ketentuan yang berlaku di Universitas Kristen Satya Wacana.

Salatiga, 26 JANUARI 2017



DIAN AYU WULANDARI



PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN PERSETUJUAN AKSES

Saya yang bertanda tangan di bawah ini:

Nama : DIAN AYU WULANDARI
NIM : 672013137 Email : AYUD570@GMAIL.COM
Fakultas : FTI Program Studi : Teknik Informatika
Judul tugas akhir : PENCARIAN SPBU TERDEKAT MENGGUNAKAN
FINITE STATE AUTOMATA (STUDI KASUS: SPBU
WILAYAH SALATIGA)

Dengan ini saya menyerahkan hak *non-eksklusif** kepada Perpustakaan Universitas – Universitas Kristen Satya Wacana untuk menyimpan, mengatur akses serta melakukan pengelolaan terhadap karya saya ini dengan mengacu pada ketentuan akses tugas akhir elektronik sebagai berikut (beri tanda pada kotak yang sesuai):

- ☒ a. Saya mengizinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA
- ☐ b. Saya tidak mengizinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA**

* Hak yang tidak terbatas hanya bagi satu pihak saja. Pengajar, peneliti, dan mahasiswa yang menyerahkan hak non-eksklusif kepada Repositori Perpustakaan Universitas saat mengumpulkan hasil karya mereka masih memiliki hak copyright atas karya tersebut.

** Hanya akan menampilkan halaman judul dan abstrak. Pilihan ini harus dilampiri dengan penjelasan/ alasan tertulis dari pembimbing TA dan diketahui oleh pimpinan fakultas (dekan/kaprodi).

Demikian pernyataan ini saya buat dengan sebenarnya.

Salatiga, 26 JANUARI 2017

Mengetahui,

MARDALENA A. INEKE PAKERENG, M.KOM

Tanda tangan & nama terang pembimbing 1

DIAN AYU WULANDARI

Tanda tangan & nama terang mahasiswa

Tanda tangan & nama terang pembimbing 2

F-LIB-081

Lembar Persetujuan

**Pencarian SPBU Terdekat
Menggunakan *Finite State Automata*
(Studi Kasus: SPBU Wilayah Salatiga)**

Artikel Ilmiah

**Peneliti :
Diah Ayu Wulandari (672013137)**

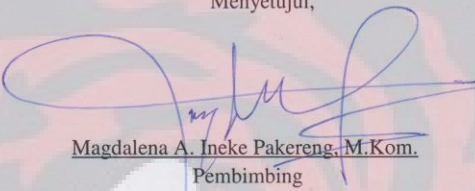
Telah disetujui untuk diuji:
Tanggal : 24 November 2016

**Magdalena A. Ineke Pakereng, M.Kom.
Pembimbing**

Lembar Pengesahan

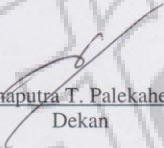
Judul Tugas Akhir : Pencarian SPBU Terdekat Menggunakan *Finite State Automata* (Studi Kasus: SPBU Wilayah Salatiga)
Nama Mahasiswa : Diah Ayu Wulandari
NIM : 672013137
Program Studi : Teknik Informatika
Fakultas : Teknologi Informasi

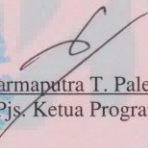
Menyetujui,


Magdalena A. Ineke Pakereng, M.Kom.

Pembimbing

Mengesahkan,

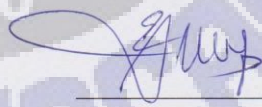

Dr. Dharmaputra T. Palekahelu, M.Pd.
Dekan


Dr. Dharmaputra T. Palekahelu, M.Pd.
Pjs. Ketua Program Studi

Dinyatakan Lulus Tanggal: 10 Januari 2017

Reviewer :

- Prof. Dr. Ir. Eko Sedyono, M.Kom.



Pencarian SPBU Terdekat Menggunakan *Finite State Automata* (Studi Kasus: SPBU Wilayah Salatiga)

Diah Ayu Wulandari¹, Magdalena A. Ineke Pakereng²

Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

Jl. Diponegoro 52-60, Salatiga 50771, Indonesia

Email: 672013137@student.uksw.edu¹, ineke.pakereng@staff.uksw.edu²

Abstract

Salatiga is a small town that has a variety of cool weather typical culinary and many tourist attractions. Salatiga is also on the route of access. Fuel Filling Station (Gas Station) is an important place in a trip. Many immigrants from outside the city find it difficult to search for Gas Station is still difficult to obtain access to information. This study was conducted to search the nearest Gas Station using Finite State Automata (FSA) with the concept of Non Deterministic Finite State Automata (N-DFA). User will read input is a path to the location of the nearest Gas Station. Finite state Automata is very appropriate in the case studies in this research. Research carried out to produce a Gas Station nearby search system that can be used as access information easier for the user.

Keywords: *Salatiga, Gas Station, Search Nearby, Finite State Automata (FSA), Non Deterministic Finite State Automata (N-DFA)*

Abstrak

Salatiga merupakan kota kecil berhawa sejuk yang memiliki beragam kuliner yang khas dan banyak tempat wisata. Salatiga juga berada pada akses jalur mudik. Stasiun Pengisian Bahan Bakar (SPBU) merupakan tempat yang penting dalam sebuah perjalanan. Banyak pendatang dari luar kota merasa sulit untuk melakukan pencarian lokasi SPBU karena akses informasi masih sulit didapatkan. Penelitian ini dilakukan untuk pencarian SPBU terdekat menggunakan *Finite State Automata* (FSA) dengan konsep *Non Deterministic Finite State Automata* (N-DFA), dimana *user* akan membaca *input* yaitu jalan untuk menuju lokasi SPBU terdekat. *Finite state automata* sangat sesuai pada studi kasus di penelitian ini. Penelitian yang dilakukan menghasilkan sistem pencarian SPBU terdekat yang dapat digunakan sebagai akses informasi yang lebih mudah untuk *user*.

Kata Kunci : *Salatiga, SPBU, Pencarian Lokasi Terdekat, Finite State Automata (FSA), Non Deterministic Finite State Automata (N-DFA)*

¹ Mahasiswa Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana Salatiga.

² Staf Pengajar Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salatiga

1. Pendahuluan

Alat transportasi merupakan hal yang penting bagi kehidupan manusia. Apabila seseorang ingin pergi atau mengunjungi suatu tempat akan menggunakan kendaraan seperti, sepeda motor atau mobil. Kendaraan pasti membutuhkan bahan bakar untuk menjalankan mesinnya. Pada saat bahan bakar hampir habis dalam perjalanan, seseorang akan mencari lokasi Stasiun Pengisian Bahan Bakar (SPBU) terdekat di sepanjang jalan untuk mengisi bahan bakar.

Salatiga adalah sebuah kota yang berhawa sejuk yang terletak di atas ketinggian sekitar 450 – 800 meter di atas air laut, dengan luas wilayah $\pm 60 \text{ km}^2$ [1]. Walaupun kota kecil Salatiga memiliki beragam kuliner yang khas dan memiliki banyak tempat wisata, sehingga banyak pengendara dari luar kota berdatangan ke Kota Salatiga. Salatiga juga berada pada akses jalur mudik yang membuat banyak pengendara dari luar kota melewati Kota Salatiga untuk menuju daerah masing-masing. SPBU merupakan tempat yang paling dibutuhkan dalam sebuah perjalanan. *User* akan mencari SPBU terdekat sebagai tempat pengisian bahan bakar dan sebagai tempat beristirahat saat lelah dalam perjalanan. Bagi orang yang mengetahui lokasi SPBU akan dengan mudah menemukan lokasi SPBU terdekat, akan tetapi bagi orang yang tidak mengetahui lokasi SPBU akan menjadi masalah dan merasa kesulitan untuk menemukan lokasi SPBU terdekat. Oleh sebab itu, diperlukan sebuah sistem untuk memudahkan pengendara dalam menemukan lokasi SPBU. Berdasarkan latar belakang yang ada, maka dilakukan penelitian yang bertujuan untuk mencari SPBU terdekat menggunakan *finite state automata*.

2. Tinjauan Pustaka

Penerapan Teori Bahasa dan Automata di Indonesia sudah sangat berkembang dan banyak diminati, terutama pada pengembangan ilmu komputer. Beberapa lembaga pendidikan atau perorangan menerapkan Teori Bahasa dan Automata untuk memecahkan suatu masalah. Salah satunya adalah untuk mengkonversikan teks menjadi suara. Pada penelitian tersebut sistem konversi teks menjadi suara tersusun atas dua sistem yaitu analisa teks dan penghasil suara percakapan. Untuk mewujudkan program konversi teks mejadi suara tersebut, pada penelitian tersebut memanfaatkan metode penggal kata *finite state automata*. Pada penelitian diperoleh aplikasi konversi teks menjadi suara dengan *input* dan *output* bahasa Indonesia [2].

Pada penelitian yang berjudul Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma *Dijkstra* (Studi Kasus Di Kabupaten Jember), membahas tentang pemetaan lokasi, pencarian lokasi, dan penentuan jalur terpendek untuk mencapai lokasi SPBU tersebut. Pencarian SPBU terdekat pada penelitian ini menggunakan algoritma *dijkstra*. Pencarian SPBU terdekat dipengaruhi oleh kriteria, *cost*, dan *reverse _cost*. Dimana untuk jalan satu arah diberikan nilai *reverse _cost* sebesar 1000000, sehingga jalan ini tidak akan pernah dipilih [3].

Berdasarkan penelitian-penelitian terdahulu terkait penerapan *finite state automata* pencarian lokasi terdekat, maka akan dilakukan penelitian yang membahas Pencarian SPBU Terdekat Menggunakan *Finite State Automata* (Studi kasus: SPBU wilayah Salatiga). Model pencarian ini menerapkan metode *Finite State Automata* untuk menentukan lokasi awal sebagai *input* dan lokasi tujuan sebagai *output* yang tentunya akan memudahkan masyarakat dalam mencari lokasi tujuan.

Finite state automata / *state* otomata merupakan suatu model matematika dari suatu sistem yang menerima *input* dan *output* diskrit. *Finite state automata* merupakan mesin otomata dari bahasa regular. Suatu *finite state automata* memiliki *state* yang banyaknya berhingga, dan dapat berpindah-pindah dari suatu *state* ke *state* lain. Secara formal *finite state automata* dinyatakan oleh 5 tupel atau $M=(Q, \Sigma, \delta, S, F)$, dimana: [4]

Q = himpunan *state* / kedudukan

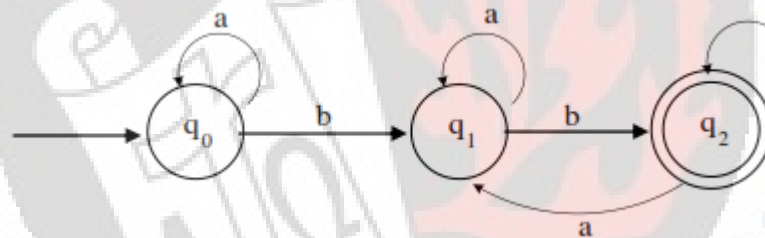
Σ = himpunan simbol *input* / masukan / abjad

δ = fungsi transisi

S = *state* awal / kedudukan awal (*initial state*)

F = himpunan *state* akhir

Sebagai contoh, sebuah otomata seperti pada Gambar 1.



Gambar 1 Diagram State [4]

Keterangan dari Gambar 1, sebagai berikut:

$Q = \{q_0, q_1, q_2\}$

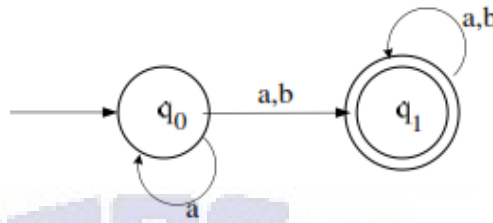
$\Sigma = \{a, b\}$

$S = q_0$

$F = \{q_1\}$

$\delta = \{ ((q_0, a), q_0), ((q_0, b), q_1), ((q_1, a), q_1), ((q_1, b), q_2), ((q_2, a), q_1), ((q_2, b), q_2) \}$

Finite state automata yang memiliki tepat satu *state* berikutnya untuk setiap simbol masukan yang diterima disebut *Deterministic Finite Automata* (DFA). Berbeda halnya dengan *Non Deterministic Finite Automata* (N-DFA) dari suatu *input* bisa dihasilkan lebih dari satu *state* berikutnya. N-DFA didefinisikan pula dengan 5 tupel. Sebagai contoh seperti pada Gambar 2.



Gambar 2 Diagram N-DFA [4]

Keterangan dari Gambar 2, sebagai berikut:

$Q = \{q_0, q_1\}$

$\Sigma = \{a, b\}$

$S = q_0$

$F = \{q_1\}$

$\delta = \{ ((q_0, a), q_0), ((q_0, a), q_1), ((q_0, b), q_1), ((q_1, a), q_1), ((q_1, b), q_1) \}$

Pada Gambar 2, apabila *state* q_0 mendapat *input* a maka bisa berpindah ke *state* q_0 atau q_1 . Jadi *Non Deterministic Finite Automata* (N-DFA) tidak pasti arahnya.

Penelitian yang dilakukan membahas tentang pencarian lokasi terdekat, dalam hal ini lokasi SPBU. SPBU (Stasiun Pengisian Bahan Bakar) merupakan tempat dimana kendaraan bermotor bisa memperoleh bahan bakar. SPBU, pada umumnya menyediakan beberapa jenis bahan bakar, misalnya: Premium, Pertamax, Solar, Pertamina Dex, Pertamax Racing, dan masih banyak lagi. Banyak Stasiun Pengisian Bahan Bakar yang juga menyediakan layanan tambahan. Misalnya, musholla, pompa angin, toilet, *minimarket*, ATM dan lain sebagainya. Stasiun Bahan Bakar juga menjadi *meeting point* atau tempat istirahat. Ada beberapa Stasiun Pengisian Bahan Bakar, terutama di jalan tol atau jalan antar kota, memiliki kedai kopi seperti Starbucks, atau restoran *fast food* dalam berbagai merek. Di beberapa negara, termasuk Indonesia, Stasiun Pengisian Bahan Bakar dijaga oleh petugas yang mengisikan bahan bakar kepada pelanggan. Pelanggan kemudian membayar biaya pengisian kepada petugas [5].

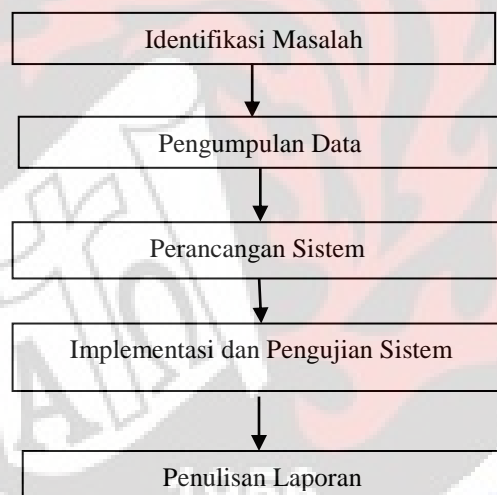
Android adalah *software* untuk perangkat mobile yang mencakup sistem operasi, *middleware* dan aplikasi kunci yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi yang digunakan oleh bermacam peranti bergerak. Karena keunggulan utama Android adalah gratis dan *open source*, yang membuat *smartphone* Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone, dan fitur (*hardware*) yang ditawarkan Android lebih baik. Beberapa fitur utama dari Android antara lain WiFi *hotspot*, *Multi-touch*, *Multitasking*, GPS, *accelerometers*, *support java*, dan mendukung banyak jaringan (GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE & WiMAX) [6].

Pada *handphone* Android terdapat fitur GPS yang dibutuhkan untuk pencarian SPBU terdekat yang menerapkan *finite state automata*. GPS merupakan sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi

secara instan, kecepatan dan informasi waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca apapun. Sedangkan alat untuk menerima sinyal satelit yang dapat digunakan oleh pengguna secara umum dinamakan *GPS Tracker* atau *GPS Tracking*. *User* dapat melacak posisi kendaraan, armada, ataupun mobil dalam keadaan *Real-Time* menggunakan GPS. Bagian yang paling penting dalam sistem navigasi GPS adalah beberapa satelit yang berada di orbit bumi atau yang sering disebut di ruang angkasa. Satelit GPS saat ini berjumlah 24 unit yang semuanya dapat memancarkan sinyal ke bumi, kemudian dapat ditangkap oleh alat penerima sinyal tersebut atau *GPS Tracker* [7].

3. Metode dan Perancangan Sistem

Penelitian yang dilakukan, diselesaikan melalui tahapan penelitian yang terbagi dalam lima tahapan, yaitu: (1) Identifikasi masalah, (2) Pengumpulan data, (3) Perancangan sistem, (4) Implementasi dan pengujian sistem, (5) Penulisan laporan.



Gambar 3 Tahapan Penelitian

Tahapan penelitian pada Gambar 3, dapat dijelaskan sebagai berikut. *Tahap pertama*: identifikasi masalah, yaitu mengidentifikasi masalah dari judul terhadap permasalahan yang ada terkait dengan proses pencarian SPBU terdekat menggunakan *finite state automata* (Studi Kasus: SPBU wilayah Salatiga); *Tahap kedua*: mengumpulkan data SPBU yang ada di wilayah Salatiga; *Tahap ketiga*: merancang sistem menggunakan *finite state automata*; *Tahap keempat*: implementasi dan pengujian sistem, yaitu membuat sistem dengan mengimplementasikan *finite state automata* untuk mencari lokasi SPBU terdekat dan dilakukan pengujian sistem yang telah dibuat, yaitu dengan menjalankan proses implementasi program serta menguji hasil *output* dari program dengan melihat hasil yang diberikan apakah sudah sesuai dengan konsep pencarian SPBU terdekat menggunakan *finite state automata*, apabila masih terdapat kesalahan dilakukan perbaikan sehingga mendapatkan hasil yang lebih

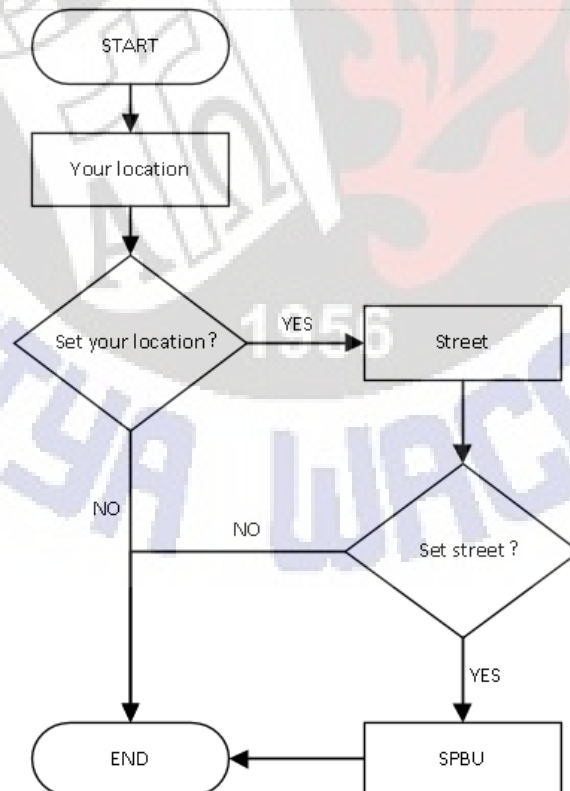
baik; *Tahap kelima*: yang dilakukan adalah menyusun laporan dari hasil pengujian yang telah dilakukan.

Pengumpulan data lokasi SPBU di wilayah Salatiga, dalam penelitian ini merupakan data sekunder yang diperoleh melalui internet dan dapat dilihat pada Tabel 1 [8] [9].

Tabel 1 Lokasi SPBU [8] [9]

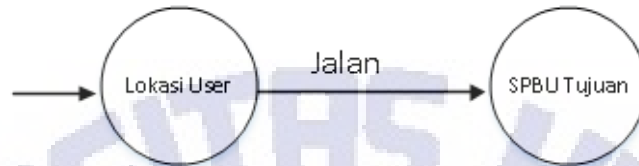
| Nama SPBU | Alamat | Latitude | Longitude |
|----------------|------------------------------|-----------|------------|
| SPBU 44.507.04 | Jl. Soekarno Hatta | -7.360828 | 110.513705 |
| SPBU 44.507.05 | Jl. Diponegoro No. 124 | -7.305765 | 110.487994 |
| SPBU 44.507.06 | Jl. Veteran | -7.337882 | 110.499156 |
| SPBU 44.507.11 | Jl. Imam Bonjol | -7.320352 | 110.494283 |
| SPBU 44.507.13 | Jl. Brigjen Sudiarto No. 5-7 | -7.332029 | 110.499150 |
| SPBU 44.507.14 | Jl. Patimura No.43 | -7.320247 | 110.503765 |
| SPBU 34.501.17 | Jl. Osamaliki | -7.329989 | 110.496019 |

Perancangan sistem yang dibangun dijelaskan dalam rancangan proses pencarian SPBU terdekat, yang ditunjukkan pada Gambar 4.



Gambar 4 Proses Pencarian SPBU Terdekat

Gambar 4 menunjukkan segala kemungkinan dari proses pencarian lokasi SPBU terdekat dalam bentuk *flowchart*. *Flowchart* ini digunakan untuk merancang diagram *state* dan alur pembuatan program.

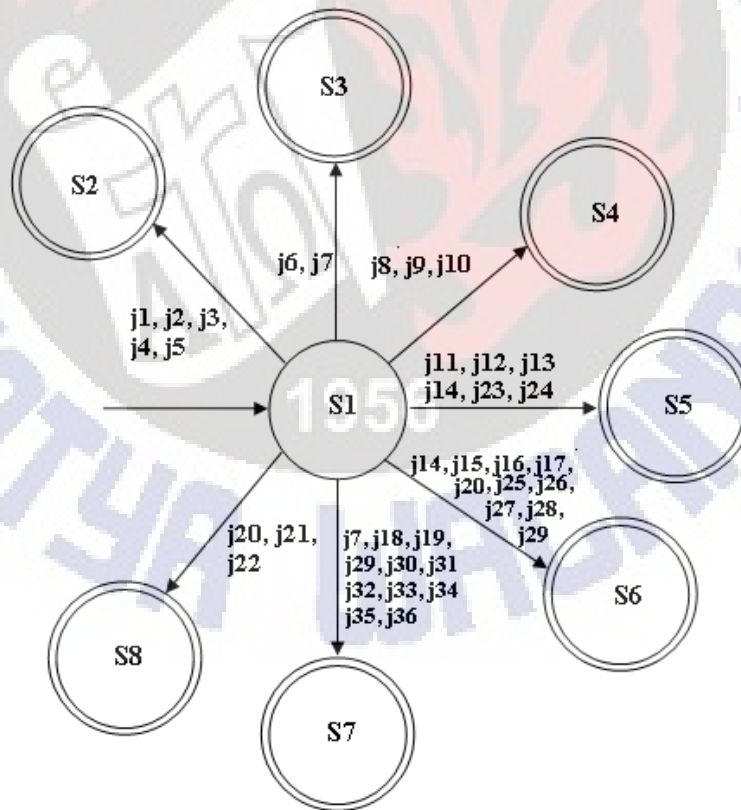


Gambar 5 Proses Diagram *State*

Gambar 5 menunjukkan proses diagram *state* dari sistem. *Start state* dari proses diagram *state* adalah lokasi *user* dan menuju ke *final state* yaitu SPBU tujuan dengan membaca *input* berupa jalan.

4. Hasil dan Pembahasan

Hasil rancangan sistem pencarian SPBU terdekat menggunakan *finite state automata* di wilayah Salatiga, dijelaskan sebagai berikut.



Gambar 6 Rancangan Diagram *State* N-DFA Pencarian SPBU Terdekat

Gambar 6 menunjukkan rancangan diagram N-DFA sistem pencarian SPBU terdekat dengan *tuple* sebagai berikut:

$Q = \{S1, S2, S3, S4, S5, S6, S7, S8\}$

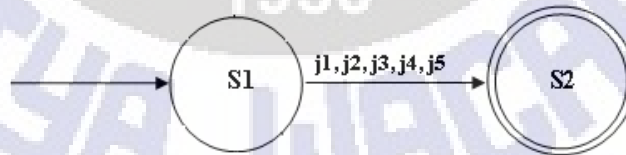
$\Sigma = \{j1, j2, j3, j4, j5, j6, j7, j8, j9, j10, j11, j12, j13, j14, j15, j16, j17, j18, j19, j20, j21, j22, j23, j24, j25, j26, j27, j28, j29, j30, j31, j32, j33, j34, j35, j36\}$

$S = S1$

$F = \{S2, S3, S4, S5, S6, S7, S8\}$

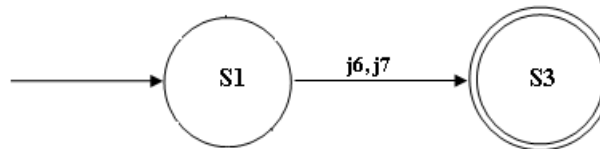
$\Delta = \{((S1, j1), S2), ((S1, j2), S2), ((S1, j3), S2), ((S1, j4), S2), ((S1, j5), S2), ((S1, j6), S3), ((S1, j7), S3), ((S1, j8), S4), ((S1, j9), S4), ((S1, j10), S4), ((S1, j11), S5), ((S1, j12), S5), ((S1, j13), S5), ((S1, j14), S5), ((S1, j23), S5), ((S1, j24), S5), ((S1, j14), S6), ((S1, j15), S6), ((S1, j16), S6), ((S1, j17), S6), ((S1, j20), S6), ((S1, j25), S6), ((S1, j26), S6), ((S1, j27), S6), ((S1, j28), S6), ((S1, j29), S6), ((S1, j7), S7), ((S1, j18), S7), ((S1, j19), S7), ((S1, j29), S7), ((S1, j30), S7), ((S1, j31), S7), ((S1, j32), S7), ((S1, j33), S7), ((S1, j34), S7), ((S1, j35), S7), ((S1, j36), S7), ((S1, j20), S8), ((S1, j21), S8), ((S1, j22), S8)\}$

Ada 8 (delapan) *state* yaitu (S1) lokasi *user*, (S2) SPBU 44.507.04, (S3) SPBU 44.507.05, (S4) SPBU 44.507.06, (S5) SPBU 44.507.11, (S6) SPBU 44.507.13, (S7) SPBU 44.507.14, (S8) SPBU 34.501.17. Sedangkan *input* yang ada sebagai berikut: (j1) Jalan Soekarno-Hatta, (j2) Jalan Tingkir, (j3) Jalan Salatiga-Solo, (j4) Jalan Lingkar Selatan, (j5) Jalan Suruh-Salatiga, (j6) Jalan Blotongan, (j7) Jalan Diponegoro, (j8) Jalan Veteran, (j9) Jalan Salatiga-Kopeng, (j10) Jalan Jend. A. Yani, (j11) Jalan Imam Bonjol, (j12) Jalan Kauman, (j13) Jalan Seruni, (j14) Jalan Kartini, (j15) Jalan Brigjen Sudiarto, (j16) Jalan Kalinongko, (j17) Jalan Jendral Sudirman, (j18) Jalan Yos Sudarso, (j19) Jalan Patimura, (j20) Jalan Tentara Pelajar, (j21) Jalan Osamaliki, (j22) Jalan Merak, (j23) Jalan K. H. Wahid Hasyim, (j24) Jalan Sinoman Tempel, (j25) Jalan Adisucipto, (j26) Jalan Sukowati, (j27) Jalan Merbabu, (j28) Jalan Pemotongan, (j29) Jalan Moh. Yamin, (j30) Jalan Raden Patah, (j31) Jalan Pemuda, (j32) Jalan Monginsidi, (j33) Jalan Buk Suling, (j34) Jalan Benoyo, (j35) Jalan Dr Muwardi, (j36) Jalan Taman Pahlawan.



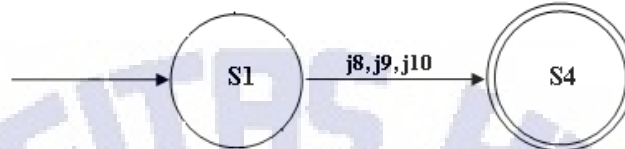
Gambar 7 Diagram State Pencarian SPBU 44.507.04

Pada Gambar 7, apabila *state* S1 berada di daerah sekitar Tingkir maka sistem akan membaca *input* j1, j2, j3, j4 atau j5 agar diterima *state* S2 yang merupakan *final state*.



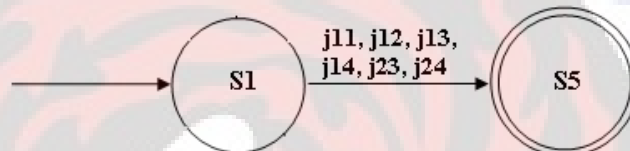
Gambar 8 Diagram State Pencarian SPBU 44.507.05

Pada Gambar 8, apabila *state* S1 berada di daerah sekitar Blotongan maka sistem akan membaca *input* j6 atau j7 agar diterima *state* S3 yang merupakan *final state*.



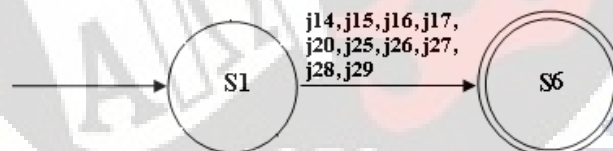
Gambar 9 Diagram State Pencarian SPBU 44.507.06

Pada Gambar 9, apabila *state* S1 berada di sekitar Jalan Veteran maka sistem akan membaca *input* j8, j9, atau j10 agar diterima *state* S4 yang merupakan *final state*.



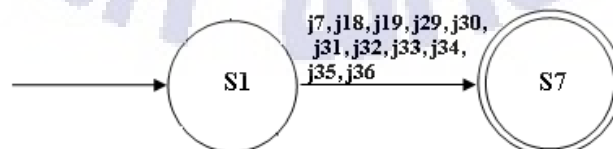
Gambar 10 Diagram State Pencarian SPBU 44.507.11

Pada Gambar 10, apabila *state* S1 berada di sekitar Jalan Imam Bonjol maka sistem akan membaca *input* j11, j12, j13, j14, j23, atau j24 agar diterima *state* S5 yang merupakan *final state*.



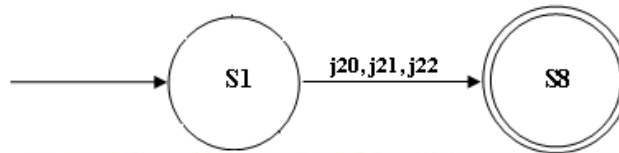
Gambar 11 Diagram State Pencarian SPBU 44.507.13

Pada Gambar 11, apabila *state* S1 berada di daerah sekitar Pancasila maka sistem akan membaca *input* j14, j15, j16, j17, j20, j25, j26, j27, j28, atau j29 agar diterima *state* S6 yang merupakan *final state*.



Gambar 12 Diagram State Pencarian SPBU 44.507.14

Pada Gambar 12, apabila *state* S1 berada di sekitar Jalan Patimura maka sistem akan membaca *input* j7, j18, j19, j29, j30, j31, j32, j33, j34, j35, atau j36 agar diterima *state* S7 yang merupakan *final state*.



Gambar 13 Diagram *State* Pencarian SPBU 44.501.17

Pada Gambar 13, apabila *state* S1 berada di sekitar Jalan Osamaliki maka sistem akan membaca *input* j20, j21, atau j22 agar diterima *state* S8 yang merupakan *final state*.

Tahap selanjutnya adalah mengaplikasikannya dalam bentuk program aplikasi. Dalam pembuatan program aplikasi diperlukan sebuah algoritma dan *pseudocode*. Algoritma pencarian SPBU terdekat didasarkan pada rancangan diagram N-DFA. Algoritma tersebut akan mempermudah pembuatan program aplikasi.

Algoritma Pencarian SPBU terdekat dapat dijelaskan sebagai berikut :

1. Tentukan lokasi user
2. Jika lokasi user berada di Jalan Soekarno-Hatta, Jalan Tingkir, Jalan Salatiga-Solo, Jalan Lingkar Selatan, atau Jalan Suruh-Salatiga maka menuju ke SPBU 44.507.04
3. Jika lokasi user berada di Jalan Blotongan atau Jalan Diponegoro maka menuju ke SPBU 44.507.05
4. Jika lokasi user berada di Jalan Veteran, Jalan Salatiga-Kopeng, Jalan Jend. A. Yani, atau Jalan Buk Suling maka menuju ke SPBU 44.507.06
5. Jika lokasi user berada di Jalan Imam Bonjol, Jalan Kauman, Jalan Seruni, Jalan Kartini, Jalan Diponegoro, Jalan K. H. Wahid Hasyim, atau Jalan Sinoman Tempel maka menuju ke SPBU 44.507.11
6. Jika lokasi user berada di Jalan Brigjen Sudiarto, Jalan Kalinongko, Jalan Jendral Sudirman, Jalan Kartini, Jalan Tentara Pelajar, Jalan Adi Sucipto, Jalan Sukowati, Jalan Merbabu, Jalan Pemotongan, atau Jalan Moh. Yamin maka menuju ke SPBU 44.507.13
7. Jika lokasi user berada di Jalan Yos Sudarso, Jalan Patimura, Jalan Diponegoro, Jalan Moh. Yamin, Jalan Raden Patah, Jalan Pemuda, Jalan Monginsidi, Jalan Buk Suling, Jalan Benoyo, Jalan Dr Muwardi, atau Jalan Taman Pahlawan maka menuju ke SPBU 44.507.14
8. Jika lokasi user berada di Jalan Osamaliki, Jalan Tentara Pelajar, atau Jalan Merak maka menuju ke SPBU 34.501.17

Pada algoritma penarian SPBU terdekat dapat diperoleh *pseudocode* yang dijelaskan sebagai berikut :

```

if lokasiuser == Jalan Soekarno-Hatta OR lokasiuser == Jalan Tingkir OR
  lokasiuser == Jalan Salatiga-Solo OR lokasiuser == Jalan Lingkar
  Selatan lokasiuser == Jalan Suruh-Salatiga
  then tampil lokasi SPBU 44.507.04
else if lokasiuser == Jalan Blotongan OR lokasiuser == Jalan Diponegoro

```



```
        then tampil lokasi SPBU 44.507.05  
else if lokasiuser == Jalan Veteran OR lokasiuser == Jalan Salatiga-Kopeng  
    OR lokasiuser == Jalan Jend. A. Yani
```



```

        then tampil lokasi SPBU 44.507.06
    else if lokasiuser == Jalan Imam Bonjol OR lokasiuser == Jalan Kauman OR
        lokasiuser == Jalan Seruni OR lokasiuser == Jalan Kartini OR
        lokasiuser == Jalan K. H. Wahid Hasyim OR
        lokasiuser == Jalan Sinoman Tempel
        then tampil lokasi SPBU 44.507.11
    else if lokasiuser == Jalan Kartini OR lokasiuser == Jalan Brigjen Sudiarto
        OR lokasiuser == Jalan Kalinongko OR lokasiuser == Jalan Jendral
        Sudirman OR lokasiuser == Jalan Tentara Pelajar OR lokasiuser ==
        Jalan Adisucipto OR lokasiuser == Jalan Sukowati OR lokasiuser ==
        Jalan Merbabu OR lokasiuser == Jalan Pemotongan OR
        lokasiuser == Jalan Moh. Yamin
        then tampil lokasi SPBU 44.507.13
    else if lokasiuser == Jalan Yos Sudarso OR lokasiuser == Jalan Patimura OR
        lokasiuser == Jalan Diponegoro OR lokasiuser == Jalan Raden Patah OR
        lokasiuser == Jalan Pemuda OR lokasiuser == Jalan Monginsidi OR
        lokasiuser == Jalan Buk Suling OR lokasiuser == Jalan Benoyo OR
        lokasiuser == Jalan Dr Muwardi OR lokasiuser == Jalan Taman Pahlawan
        then tampil lokasi SPBU 44.507.14
    else if lokasiuser == Jalan Tentara Pelajar OR lokasiuser == Jalan Osamiliki
        OR lokasiuser == Jalan Merak
        then tampil lokasi SPBU 34.501.17
    else
        tampil Inputan Salah

```

Pseudocode pencarian SPBU terdekat digunakan sebagai dasar pembuatan program aplikasi. Perintah yang dibuat berdasarkan *pseudocode*, ditunjukkan pada Kode Program 1.

Kode Program 1 Perintah untuk Pencarian Lokasi SPBU

```

1. if(myLocation.getText().toString().equals("jalan soekarno hatta"))||
2.     myLocation.getText().toString().equals("tingkir") ||
3.     myLocation.getText().toString().equals("jalan salatiga-solo"))||
4.     myLocation.getText().toString().equals("jalan lingkar selatan"))||
5.     myLocation.getText().toString().equals("jalan suruh-salatiga")){
6.         LatLng spbuTingkir = new LatLng(-7.360828, 110.513705);
7.         MarkerOptions markerSpbul = new MarkerOptions();
8.         markerSpbul.position(spbuTingkir); markerSpbul.title("SPBU 44.507.04");
9.         markerSpbul.snippet("SPBU 44.507.04 Tingkir");
10.        markerSpbul.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorF
actory.HUE_ROSE)); spbu.addMarker(markerSpbul);
11.        spbu.animateCamera(CameraUpdateFactory.newLatLngZoom(spbuTingkir,15));
12. }else if(myLocation.getText().toString().equals("jalan diponegoro"))||
13.     myLocation.getText().toString().equals("blotongan")){
14.         LatLng spbuDipo = new LatLng(-7.305765, 110.487994);
15.         MarkerOptions markerSpbu2 = new MarkerOptions();
16.         markerSpbu2.position(spbuDipo); markerSpbu2.title("SPBU 44.507.05");
17.         markerSpbu2.snippet("SPBU 44.507.05 Soka");
18.         markerSpbu2.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorF
actory.HUE_ROSE)); spbu.addMarker(markerSpbu2);
19.        spbu.animateCamera(CameraUpdateFactory.newLatLngZoom(spbuDipo,15));
20. }else if(myLocation.getText().toString().equals("jalan veteran"))||
21.     myLocation.getText().toString().equals("jalan salatiga-kopeng") ||
22.     myLocation.getText().toString().equals("jalan a yani")){
23.         LatLng spbuVeteran = new LatLng(-7.337882, 110.499156);
24.         MarkerOptions markerSpbu3 = new MarkerOptions();
25.         markerSpbu3.position(spbuVeteran); markerSpbu3.title("SPBU 44.507.06");
26.         markerSpbu3.snippet("SPBU 44.507.06 Veteran");
27.         markerSpbu3.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorF
actory.HUE_ROSE));

```

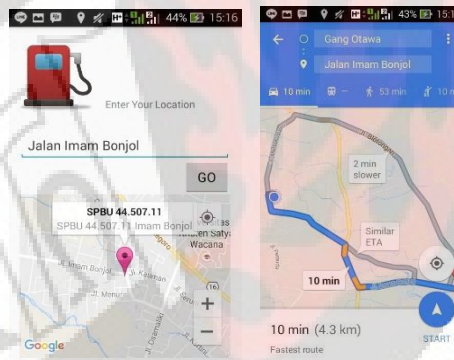
```

28.         spbu.addMarker(markerSpu3);
29.         spbu.animateCamera(CameraUpdateFactory.newLatLngZoom(spbuVeteran,15));
30. }else if(myLocation.getText().toString().equals("jalan imam bonjol")){
31.     myLocation.getText().toString().equals("kauman") ||
32.     myLocation.getText().toString().equals("seruni") ||
33.     myLocation.getText().toString().equals("jalan kartini") ||
34.     myLocation.getText().toString().equals("jalan wahid hasyim") ||
35.     myLocation.getText().toString().equals("jalan sinoman tempel")){
36.         LatLng spbuBonjol = new LatLng(-7.320352, 110.494283);
37.         MarkerOptions markerSpu4 = new MarkerOptions();
38.         markerSpu4.position(spbuBonjol); markerSpu4.title("SPBU 44.507.11");
39.         markerSpu4.snippet("SPBU 44.507.11 Jetis");
40.         markerSpu4.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ROSE)); spbu.addMarker(markerSpu4);
41.         spbu.animateCamera(CameraUpdateFactory.newLatLngZoom(spbuBonjol,15));
42. }else if(myLocation.getText().toString().equals("jalan birgjen sudiarto")){
43.     myLocation.getText().toString().equals("kalinongko") ||
44.     myLocation.getText().toString().equals("jalan jendral sudirman") ||
45.     myLocation.getText().toString().equals("jalan kartini") ||
46.     myLocation.getText().toString().equals("jalan tentara pelajar") ||
47.     myLocation.getText().toString().equals("jalan adisucipto") ||
48.     myLocation.getText().toString().equals("jalan sukowati") ||
49.     myLocation.getText().toString().equals("jalan merbabu") ||
50.     myLocation.getText().toString().equals("jalan pemotongan") ||
51.     myLocation.getText().toString().equals("jalan moh yamin")){
52.         LatLng spbuPancasila = new LatLng(-7.332029, 110.499150);
53.         MarkerOptions markerSpu5 = new MarkerOptions();
54.         markerSpu5.position(spbuPancasila); markerSpu5.title("SPBU 44.507.13");
55.         markerSpu5.snippet("SPBU 44.507.13 Pancasila");
56.         markerSpu5.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ROSE)); spbu.addMarker(markerSpu5);
57.         spbu.animateCamera(CameraUpdateFactory.newLatLngZoom(spbuPancasila,15));
58. }else if(myLocation.getText().toString().equals("jalan yos sudarso")){
59.     myLocation.getText().toString().equals("jalan patimura") ||
60.     myLocation.getText().toString().equals("jalan diponegoro") ||
61.     myLocation.getText().toString().equals("jalan raden patah") ||
62.     myLocation.getText().toString().equals("jalan pemuda") ||
63.     myLocation.getText().toString().equals("monginsidi") ||
64.     myLocation.getText().toString().equals("jalan buk suling") ||
65.     myLocation.getText().toString().equals("jalan benoyo") ||
66.     myLocation.getText().toString().equals("jalan dr muwardi") ||
67.     myLocation.getText().toString().equals("taman pahlawan") ||
68.     myLocation.getText().toString().equals("jalan moh yamin")){
69.         LatLng spbuPatimura = new LatLng(-7.320247, 110.503765);
70.         MarkerOptions markerSpu6 = new MarkerOptions();
71.         markerSpu6.position(spbuPatimura); markerSpu6.title("SPBU 44.507.14");
72.         markerSpu6.snippet("SPBU 44.507.14 Patimura");
73.         markerSpu6.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ROSE)); spbu.addMarker(markerSpu6);
74.         spbu.animateCamera(CameraUpdateFactory.newLatLngZoom(spbuPatimura,15));
75. }else if(myLocation.getText().toString().equals("jalan osamaliki")){
76.     myLocation.getText().toString().equals("jalan merak") ||
77.     myLocation.getText().toString().equals("jalan tentara pelajar")){
78.         LatLng spbuOsamaliki = new LatLng(-7.329989, 110.496019);
79.         MarkerOptions markerSpu7 = new MarkerOptions();
80.         markerSpu7.position(spbuOsamaliki); markerSpu7.title("SPBU 44.501.17");
81.         markerSpu7.snippet("SPBU 44.501.17 Osamaliki");
82.         markerSpu7.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ROSE)); spbu.addMarker(markerSpu7);
83.         spbu.animateCamera(CameraUpdateFactory.newLatLngZoom(spbuOsamaliki,15));
84. }else{
85.     Toast.makeText(getApplicationContext(), "Inputan Salah atau Belum Ada Inputan!", Toast.LENGTH_LONG).show();
86. }

```

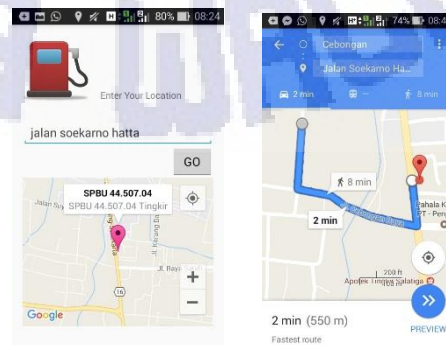
Kode Program 1 merupakan perintah untuk mencari lokasi SPBU terdekat berdasarkan lokasi *user*. Perintah yang digunakan untuk menampilkan lokasi SPBU adalah *if* dan *else*. Perintah pada baris 1 sampai baris 5, baris 12 dan baris 13, baris 20 sampai baris 22, baris 30 sampai baris 35, baris 42 sampai baris 51, baris 58 sampai baris 68, dan baris 75 sampai baris 77 merupakan perintah untuk menentukan lokasi *user*. Perintah untuk menampilkan lokasi SPBU yang merupakan *final state* atau kondisi akhir berada pada baris 6 sampai baris 11, baris 14 sampai baris 19, baris 23 sampai baris 29, baris 36 sampai baris 41, baris 52 sampai baris 57, baris 69 sampai baris 74, dan baris 78 sampai baris 83. Kondisi *else* pada baris 84 sampai baris 86 terjadi karena *user* salah *input* atau tidak memasukkan lokasi *user*.

Aplikasi dibangun menggunakan *Eclipse* dengan bahasa pemrograman Java. Program aplikasi dirancang berdasarkan hasil dari rancangan diagram N-DFA dengan menggunakan perintah *if* dan *else*. Hasil dari aplikasi akan menentukan *final state* yang dituju. Lokasi *user* diketikan secara manual dan *input* dari *user* akan diproses, sehingga menampilkan lokasi SPBU terdekat dengan lokasi *user*. Apabila *marker* lokasi SPBU diklik, akan menampilkan *direction route* yang langsung terhubung dengan GPS pada *handphone*. Tampilan hasil *output* dari program dapat dijelaskan sebagai berikut.



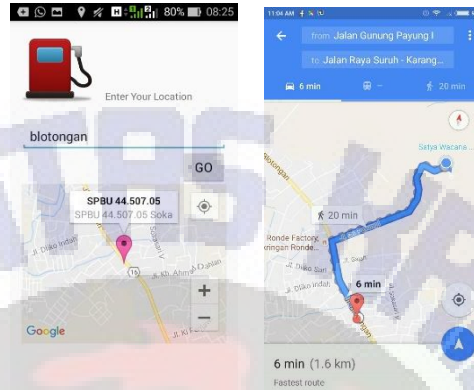
Gambar 14 Tampilan *Input* Lokasi *User* dan Tampilan *Direction Route*

Gambar 14 merupakan tampilan hasil *output* lokasi SPBU 44.507.11 yang membaca *input* dari Jalan Imam Bonjol.



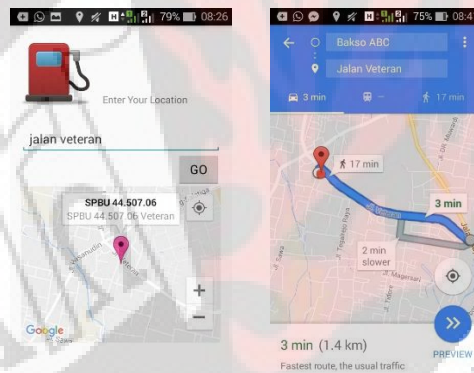
Gambar 15 Tampilan *Input* Lokasi *User* dan Tampilan *Direction Route*

Gambar 15 merupakan tampilan hasil *output* lokasi SPBU 44.507.04 yang membaca *input* dari Jalan Soekarno Hatta.



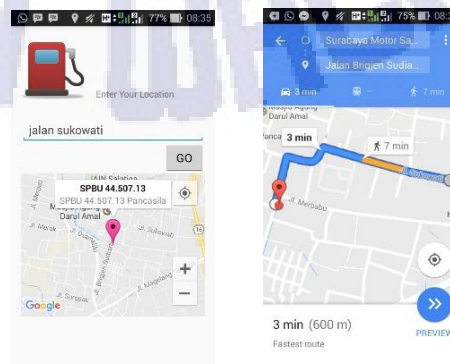
Gambar 16 Tampilan *Input* Lokasi *User* dan Tampilan *Direction Route*

Gambar 16 merupakan tampilan hasil *output* lokasi SPBU 44.507.05 yang membaca *input* dari Jalan Blotongan.



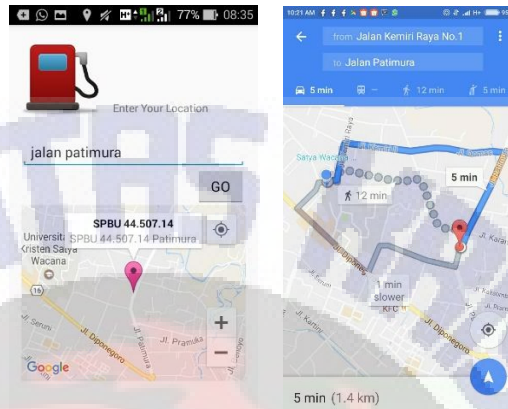
Gambar 17 Tampilan *Input* Lokasi *User* dan Tampilan *Direction Route*

Gambar 17 merupakan tampilan hasil *output* lokasi SPBU 44.507.06 yang membaca *input* dari Jalan Veteran.



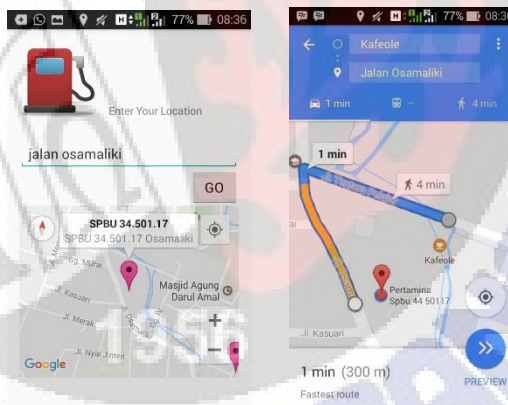
Gambar 18 Tampilan *Input* Lokasi *User* dan Tampilan *Direction Route*

Gambar 18 merupakan tampilan hasil *output* lokasi SPBU 44.507.13 yang membaca *input* dari Jalan Sukowati.



Gambar 19 Tampilan *Input* Lokasi *User* dan Tampilan *Direction Route*

Gambar 19 merupakan tampilan hasil *output* lokasi SPBU 44.507.14 yang membaca *input* dari Jalan Patimura.



Gambar 20 Tampilan *Input* Lokasi *User* dan Tampilan *Direction Route*

Gambar 20 merupakan tampilan hasil *output* lokasi SPBU 44.501.17 yang membaca *input* dari Jalan Osamaliki.

Hasil pengujian dari implementasi sistem pencarian lokasi SPBU terdekat menggunakan *Finite State Automata* dilakukan dengan membandingkan jarak antara lokasi *user* menuju ke SPBU tujuan. Jarak diperoleh dari *direction route* pada GPS. Pengujian dilakukan dengan mengambil sampel dari lokasi *user* di daerah Imam Bonjol, lokasi *user* di daerah Sukowati, dan lokasi *user* di daerah Blotongan.

Tabel 2 Hasil Pengujian di Daerah Imam Bonjol

| Nama SPBU | Jarak | Hasil Pengujian |
|----------------|--------|-------------------|
| SPBU 44.507.04 | 11 km | Bukan SPBU tujuan |
| SPBU 44.507.05 | 4,3 km | Bukan SPBU tujuan |
| SPBU 44.507.06 | 6,6 km | Bukan SPBU tujuan |
| SPBU 44.507.11 | 4,0 km | SPBU tujuan |
| SPBU 44.507.13 | 5,7 km | Bukan SPBU tujuan |
| SPBU 44.507.14 | 8,3 km | Bukan SPBU tujuan |
| SPBU 34.501.17 | 5,2 km | Bukan SPBU tujuan |

Hasil perbandingan antara jarak lokasi *user* di daerah Imam Bonjol dengan lokasi SPBU yang berada di wilayah Salatiga pada Tabel 2, disimpulkan bahwa SPBU 44.507.11 merupakan SPBU tujuan. Hasil dari pengujian ini sesuai dengan implementasi sistem dari aplikasi yang sudah dibuat.

Tabel 3 Hasil Pengujian di Daerah Sukowati

| Nama SPBU | Jarak | Hasil Pengujian |
|----------------|--------|-------------------|
| SPBU 44.507.04 | 3,9 km | Bukan SPBU tujuan |
| SPBU 44.507.05 | 4,0 km | Bukan SPBU tujuan |
| SPBU 44.507.06 | 1,7 km | Bukan SPBU tujuan |
| SPBU 44.507.11 | 2,1 km | Bukan SPBU tujuan |
| SPBU 44.507.13 | 0,6 km | SPBU tujuan |
| SPBU 44.507.14 | 1,8 km | Bukan SPBU tujuan |
| SPBU 34.501.17 | 1,1 km | Bukan SPBU tujuan |

Hasil perbandingan antara jarak lokasi *user* di daerah Sukowati dengan lokasi SPBU yang berada di wilayah Salatiga pada Tabel 3, disimpulkan bahwa SPBU 44.507.13 merupakan SPBU tujuan. Hasil dari pengujian ini sesuai dengan implementasi sistem dari aplikasi yang sudah dibuat.

Tabel 4 Hasil Pengujian di Daerah Blotongan

| Nama SPBU | Jarak | Hasil Pengujian |
|----------------|--------|-------------------|
| SPBU 44.507.04 | 8,1 km | Bukan SPBU tujuan |
| SPBU 44.507.05 | 1,6 km | SPBU tujuan |
| SPBU 44.507.06 | 5,5 km | Bukan SPBU tujuan |
| SPBU 44.507.11 | 3,0 km | Bukan SPBU tujuan |
| SPBU 44.507.13 | 4,1 km | Bukan SPBU tujuan |
| SPBU 44.507.14 | 3,8 km | Bukan SPBU tujuan |
| SPBU 34.501.17 | 4,1 km | Bukan SPBU tujuan |

Hasil perbandingan antara jarak lokasi *user* di daerah Blotongan dengan lokasi SPBU yang berada di wilayah Salatiga pada Tabel 4, disimpulkan bahwa SPBU 44.507.05 merupakan SPBU tujuan. Hasil dari pengujian ini sesuai dengan implementasi sistem dari aplikasi yang sudah dibuat.

5. Simpulan

Berdasarkan hasil penelitian dan pembahasan yang dilakukan dapat diambil kesimpulan sebagai berikut : (1) *Finite state automata* dapat diimplementasikan dalam pencarian lokasi terdekat; (2) Rancangan diagram *state* berdasarkan konsep N-DFA yang telah dibuat dapat membantu pengembangan teknologi GPS; dan (3) Pencarian lokasi SPBU terdekat membantu *user* dalam pencarian lokasi SPBU dengan mudah dan cepat. Saran yang dapat diberikan untuk pengembangan penelitian lebih lanjut sebagai berikut: (1) Penentuan lokasi *user* yang otomatis dan langsung tampil *direction route* karena dalam aplikasi yang dibangun masih secara manual; (2) Penambahan lokasi tempat penting selain SPBU dan penambahan data lokasi SPBU selain di wilayah Salatiga.

6. Daftar Pustaka

- [1] Kota Salatiga. <http://www.jatengprov.go.id/id/profil/kota-salatiga>. Diakses Tanggal 8 Agustus 2016.
- [2] Isroka, 2013. "Aplikasi Konversi Teks Menjadi Suara dengan Menggunakan Metode Penggal Kata *Finite State Automata* (FSA)". <http://jurnal-online.um.ac.id/data/artikel/artikelA6A450B00B1E3E30227F1860F1C962ED.pdf>. Diakses Tanggal 8 Agustus 2016.
- [3] Yulia, W. E., dkk., 2015. Jember: Universitas Jember. <http://jnte.ft.unand.ac.id/index.php/jnte/article/viewFile/132/139>. Diakses Tanggal 8 Agustus 2016.
- [4] Utdirartatmo, F., 2001. *Teori Bahasa dan Otomata*. Yogyakarta: Penerbit JJ Learning.
- [5] KontraktorSPBU, 2013. Stasiun Pengisian Bahan Bakar Umum (SPBU). <http://www.kontraktorspbu.com/stasiun-pengisian-bahan-bakar-umum-spbu/>. Diakses Tanggal 8 Agustus 2016.
- [6] Mandalamaya, 2015. Pengertian GPS, Cara Kerja GPS, dan Fungsi GPS. <http://www.mandalamaya.com/pengertian-gps-cara-kerja-gps-dan-fungsi-gps/>. Diakses Tanggal 8 Agustus 2016.
- [7] InfoTeknologi, 2011. Apa itu Android. <http://www.infoteknologi.com/selular/apa-itu-android>. Diakses pada Tanggal 27 Agustus 2016
- [8] Autofresh, 2015. Info Mudik. <http://autofresh.net/info-mudik/jalur-selatan.php?kota=Salatiga>. Diakses Tanggal 27 Agustus 2016.

- [9] LewatMana, 2014. Semua SPBU Pertamina (Pasti Pas) di Salatiga. <https://lewatmana.com/lokasi/spbu/spbu-pertamina/di/salatiga/>. Diakses Tanggal 27 Agustus 2016.

